C. REMARKS

Applicants respectfully request reconsideration of the outstanding rejections and reexamination of the present application in light of the following amendments and remarks.

Status of the Claims

Claims 40-58 are pending in the application. Claim 42 is amended.

Alleged Obviousness under 35 USC 103(a)

The Office Action rejects claims 40-58 under 35 USC 103(a) as being unpatenable over Kelkar (US Patent 7,058,846) in view of Franckowiak (US Publication 2003/0005356). [Office Action, p. 2] Applicants traverse the rejections of claims 40-58 and respectfully submit that the claims are not obvious as discussed below.

"Section 103 forbids issuance of a patent when 'the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.'" *KSR Int'l Co. v. Teleflex Inc.,* 127 S.Ct. 1727, 1734 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co.,* 383 U.S. 1, 17-18 (1966).

The Examiner bears the initial burden of supporting any prima facie conclusion of obviousness. See *In re Rinehart*, 531, F.2d 1048, 189, USPQ 143 (CCPA 1976); *KSR International Co. v. Teleflex Inc.,* 82 USPQ2d 1385, 1396 (2007); MPEP 2142. The key to supporting a rejection under 35 USC 103 is the clear articulation of the reasons why the claimed invention would have been obvious; the analysis supporting a rejection under 35 USC 103 should be made explicit. *See KSR International Co.*, 82 USPQ2d at 1396; MPEP 2142 (Rev. 6, Sept. 2007).

Applicants respectfully submit that when a proper Graham inquiry is made a prima facie case of obviousness is not established and the claims should be allowed.

Claims 40

executing the software application on a primary node to form a master application

identifying resources and dependencies required by the master application to form required resources

First, Applicants respectfully submit that a prima facie case of obviousness has not been established under § 103(a) because the Office Action has not established that Kelkar or Franckonwiak, separately or in combination, teach executing the software application on a primary node to form a master application *or* identifying resources and dependencies required by the master application to form required resources recited in claim 40.

In a Graham inquiry, as to the scope and contents of Kelkar, the Office Action cites col. 4, lines 49-53 of Kelkar as reading on the claimed elements of executing the software application on a primary node to form a master application and identifying resources and dependencies required by the master application to form required resources. [Office Action, p. 2] Col. 4, lines 49-53 of Kelkar read:

> One problem with system 100 described above is that storage resource definition 140D is stored on node 110A. If node 110A fails, storage resource 140 cannot be used because storage resource definition 140D is not accessible to other nodes. To make resource configuration available to another node that can resume operation of node 110A upon failure, the invention synchronizes resource configuration data on multiple nodes in a clustering environment.

Applicants note that in considering Kelkar as a whole, the storage resource definition 140D of Kelkar "defines attributes of storage resource 140 and is used by storage access interface 102A to perform storage operations." *Kelkar*, col. 4, lines 33-36. Thus, the synchronized resource configuration data described in Kelkar refers to configuration data for storage systems, not for software applications. The Examiner does not point to any portion of Kelkar which teaches executing a software application on a primary node to form *a master application*. In addition, Kelkar's description of synchronizing storage resource configuration data on multiple nodes in a

clustering environment is not the same as *identifying* resources *and dependencies* required *by the master application* to form required resources.

In addition, in a Graham inquiry, as to the scope and contents of Franckonwiak, Applicants note that the Examiner does not recite any portion of Franckonwiak as reading on executing the software application on a primary node to form a master application or identifying resources and dependencies required by the master application to form required resources.

In a Graham inquiry, in considering the differences between Kelkar and Franckonwiak and claim 40, the Office Action does not identify teachings of Kelkar that teach executing the software application on a primary node to form a master application. In addition, in considering the differences between Kelkar and claim 40, it is clear that neither of the references teaches identifying resources and dependencies required by the master application to form required resources. In particular, Kelkar's description of synchronizing storage resource configuration data on multiple nodes in a clustering environment does not teach any *identification* of resources or dependencies required by *a master application* executing on a primary node. In addition, no portion of Franckonwiak teaches identifying resources or dependencies required by a master application. Further, there is not a combination of Kelkar and Franckonwiak proposed by the Examiner that would teach identifying of resources and dependencies required by a master application.

In view of the lack of teaching in Kelkar or Franckonwiak of the claimed elements of a master application or identifying resources and dependencies required by the master application to form required resources, the Office Action does not provide a clear articulation of why claim 40 would be obvious in view of Kelkar and Franckonwiak and it is clear the Office Action has failed to establish a prima facie case of obviousness as to claim 40.

generating a structure of the master application and a dynamic graph of the required resources from the required resources;

replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica

Applicants agree with the Examiner that Kelkar does not teach the elements of updating the required resources dynamically on the primary node; generating a structure of the master

application and a dynamic graph of the required resources from the required resources; replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica; wherein the set of secondary nodes comprises one or more secondary nodes; restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications; executing the set of clone software applications on the set of secondary nodes, without loss of context; and updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application, however Applicants respectfully submit that Franckonwiak also does not teach at least one of these claim elements and therefore the Office Action fails to establish a prima facie case of obviousness as to claim 40. [Office Action, p. 3]

In a Graham inquiry, as to the scope and contents of Franckonwiak, the Office Action cites paragraph 0042 of Franckonwiak as reading on the claim element of updating the required resources dynamically on the primary node. [Office Action, p. 3] Paragraph 0042 of Franckonwiak reads:

> At regular intervals, as determined by the active application 14, when the changes to the application data are complete and consistent as shown at 58, the application invokes a logicalPush as shown at 60. Data consistency 58 is determined by the application 14 in any suitable known manner. The logicalPush 60 encodes the changed data for a group of regions into transport payloads and requests that the active DML instance 10 transfer the accumulated changes to the standby process.

In addition, as to the scope and contents of Franckonwiak, the Office Action cites Figure 2, item 56 as reading on the claim element of generating a structure of the master application and a dynamic graph of the required resources from the required resources. [Office Action, p. 3] Element 56 on Figure 2 in Franckonwiak reads "Datamarking changed tuples". Applicants note that paragraph 0041 of Franckonwiak describes "When the active application 14 changes data within the regions of the active processor memory 18, as shown at step 54, the changes to the tuple(s) are noted by a dataMark interface at 56." Further, the Office Action cites Figure 2, elements 56, 60, 68, 70, 72, and 74 of Franckonwiak as describing "This figure clearly shows that whatever takes place in the first node in the system will mirror that in the second node" and as reading on the claim elements of replicating the resources by transferring the structure to a set

of secondary nodes via a network to form a replica; wherein the set of secondary nodes comprises one or more secondary nodes; restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications; executing the set of clone software applications on the set of secondary nodes, without loss of context; and updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application. [Office Action, p. 3] In Figure 2 of Franckonwiak, element 60 reads "encoding into transport payloads (logical push)", element 68 reads "invoking application specific transport", element 70 reads "transferring data to standby process (physical push)", element 72 reads "standby processor memory receives changed data", and element 74 reads "return acknowledgment to active process."

In further considering the scope and contents of Franckonwiak beyond the portions cited by the Examiner, Applicants note that paragraph 0030 of Franckonwiak states "The invention treats the application data as blocks of data within regions of memory, thereby providing the ability to replicate the data without having to know the structure of the data."

In considering the differences between Kelkar and Franckonwiak and claim 40, Applicants respectfully submit that neither reference, separately or in combination, teaches the claimed elements of generating a structure of the master application or replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica. First, element 56 of Franckonwiak of "Datamarking changed tuples" is not the same as *generating a structure* of the master application. Second, none of the elements of Figure 2 of Franckonwiak which describe "encoding into transport payloads", "invoking application specific transport" or "transferring data to standby process" are the same as *transferring a structure* of the master application to a set of secondary nodes. Third, in considering the differences between Franckonwiak and claim 40, Franckonwiak when considered as a whole, including paragraph 0030, specifically describes replication of data without knowing the structure of the data, which teaches away from both the claimed elements of generating a *structure* of the master application and replicating resources by transferring a *structure* to a set of secondary nodes. ("A prior art reference must be considered in its entirey, i.e. as a whole, including portions that would lead

away from the claimed invention." *W.L. Gore & Associates, Inc. v. Garlock, Inc.,* 721 F.2d 1540, 220 USPQ 303 (Fed Cir. 1983), *cert denied,* 469 U.S. 851 (1984).)

Applicants respectfully submit that in view of the scope and contents of Franckonwiak and the differences between Franckonwiak and the claimed elements of generating a structure of the master application or replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica, including Franckonwiak clearly teaching away from generating *a structure* of an application and transferring *a structure*, it is clear that the differences between Kelkar and Franckonwiak and claim 40 are not such that claim 40 as a whole would have been obvious to one with skill in the art at the time of the invention.

In addition, Applicants note that the rationale stated for modifying Kelkar by Franckonwiak is that it would have been obvious to one of ordinary skill in the art to incorporate updating the required resources dynamically on the primary node; generating a structure of the master application and a dynamic graph of the required resources from the required resources; replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica; wherein the set of secondary nodes comprises one or more secondary nodes; restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications; executing the set of clone software applications on the set of secondary nodes, without loss of context; and updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application of Franckonwiak into the method for replicating a software application of Kelkar and that there would be motivation to apply Frankonwiak to Kelkar because Franckonwiak discloses "in an effort to achieve high availability, RCS application processors are paired to form mated processor pairs in an active/standby arrangement. When a fault occurs on the active processor, the standby process is elevated to the active role to continue providing service." [Office Action, pp. 4-5] Applicants respectfully note that no portion of Franckonwiak is cited to support the Examiner's conclusion as to what is disclosed in Frankonwiak. In addition, Applicants respectfully submit that the Examiner errs in characterizing Kelkar as describing a "method for replicating a software application" because it is clear that Kelkar only describes a cluster failover

management for storage resources, not for software applications. *Kelkar*, abstract, column 4, lines 53-56. Moreover, Applicants note that rejections on obviousness cannot be sustained by mere conclusory statements; instead there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *KSR International Co. v. Teleflex Inc.,* 82 USPQ2d 1385, 1396 (2007); MPEP 2141. Applicants respectfully submit that the Examiner appears to be stating a rationale of "applying" Franckonwiak to Kelkar, however Applicants respectfully submit that the Examiner has not articulated what "known technique" that one of ordinary skill in the art would have recognized that applying the known technique would have yielded predictable results and resulted in an improved system as required under MPEP 2143(D).

Therefore, because a proper Graham factual finding indicates differences between Kelkar and Franckonwiak and claim 40 and no clear articulation of the reasons why the claimed invention of claim 40 would have been obvious is provided, the Office erred in finding prima facie obviousness as to claim 1. MPEP 2141, IV. Because the Office fails to find prima facie obviousness as to claim 40, Applicants respectfully request withdrawal of the rejection under 35 USC 103(a) and allowance of the claim.

Claim 58

Claim 58 is rejected on the same grounds as claim 40. [Office Action, pp. 2-4] Applicants respectfully submit that prima facie obviousness is not established for claim 58 for at least the same reasons that prima facie obviousness is not established for claim 40. Because the Office fails to establish prima facie obviousness as to claim 58, Applicants respectfully request withdrawal of the rejection under 35 USC 103(a) and allowance of the claim.

Claims 41-57

Applicants respectfully assert that because claim 40 is nonobvious under 35 USC 103(a), claims 41-57 which depend on claim 40 are also nonobvious and should be allowed. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Claim 41

In addition, claim 41 is not obvious under 35 USC 103(a) in view of Kelkar and Franckonwiak.

As to claim 41, in a Graham inquiry, as to the scope and contents of Kelkar, the Office Action cites col. 4, lines 53-56 of Kelkar as reading on the claim element of creating and maintaining a dependency tree, based on the dynamic graph, supplying, at all times, information on the replicated resources. [Office Action, p. 5] Col. 4, lines 53-56 of Kelkar read:
To make resource configuration available to another node that can resume operation of node 110A upon failure, the invention synchronizes resource configuration data on multiple nodes in a clustering environment.

In a Graham inquiry, as to the differences between Kelkar and claim 41, Applicants respectfully submit that Kelkar does not teach the elements of claim 41 because the synchronizing of data storage resource configuration data on multiple nodes described in Kelkar is not the same as creating and maintaining *a dependency tree*, based on the *dynamic graph* of the required resources for an application. The Office Action does not cite any portion of Franckonwiak as reading on claim 41.

Therefore, because Kelkar and Franckonwiak are devoid of any teaching of creating or maintaining a dependency tree based on a dynamic graph and the Office Action fails to provide any other basis on which one of ordinary skill in the art would find claim 41 obvious in view of the differences between Kelkar and Franckonwiak and claim 41, the Office Action does not establish prima facie obviousness as to claim 41. Because the Office fails to find prima facie obviousness as to claim 41, Applicants respectfully request withdrawal of the rejection under 35 USC 103(a) and allowance of the claim.

Claim 42

As to claim 42, Applicants note that claim 42 is amended as follows and that no new matter is added through the amendments because the amendments are supported in paragraphs 0056-0059, 0067, 0072, 0078-0080 of the specification of the present application:

checkpointing the resources on the set of secondary nodes, wherein the replicated resources on the set of secondary nodes comprise a virtual memory and

calling stack of each process of the master application, at least one system resource used by the master application, and data written on disks used by the master application, wherein the checkpointing ~~having~~ has an adjustable period to optimize the difference between the recovery time after switching to the set of secondary nodes and the quantity of information to be captured and transferred to the set of secondary nodes.

In addition, Applicants respectfully submit that claim 42 is not obvious under Kelkar and Franckonwiak. Applicants submit that neither Kelkar nor Franckonwiak teach the replicated resources on the set of secondary nodes comprising each of a virtual memory and calling stack of each process of the master application, at least one system resource used by the master application, and data written on disks used by the master application. Applicants respectfully submit that in view of the lack of teaching by Kelkar and Franckonwiak of at least one element of claim 42, claim 42 is not obvious under the references and should be allowed.

Claim 47

As to claim 47, Applicants note that claim 47 is amended as follows and that now new matter is added through the amendments because the amendments are supported in paragraphs 0072-0080 of the specification of the present application:

Claim 47 (Currently Amended): The computer implemented method according to claim 42, wherein the checkpointing further comprises ~~at least one of the following~~:

processing a synchronization barrier for locking the processes of the master application to take a non-blurred photograph of the state of the primary node and the master application at each of a plurality of phases of the checkpointing;

managing resources necessary for replicating the master application during the plurality of phases of the checkpointing;

managing the at least one system resource[s] used by the master application during the plurality of phases of the checkpointing; and

managing process resources.

In addition, Applicants respectfully submit that claim 47 is not obvious under Kelkar and Franckonwiak. Applicants submit that neither Kelkar nor Franckonwiak teach checkpointing further comprising processing a synchronization barrier for locking the processes of the master application to take a non-blurred photograph of the state of the primary node and the master

application; managing resources necessary for replicating the master application during a plurality of phases of the checkpointing; managing the at least one system resource used by the master application during the plurality of phases of the checkpointing. Applicants respectfully submit that in view of the lack of teaching by Kelkar and Franckonwiak of at least one element of claim 47, claim 47 is not obvious under the references and should be allowed.

Conclusion

Applicants note the citation of pertinent prior art cited by the Examiner.

In view of the foregoing, withdrawal of the rejections and the allowance of the current pending claims is respectfully requested. If the Examiner feels that the pending claims could be allowed with minor changes, the Examiner is invited to telephone the undersigned to discuss an Examiner's Amendment.

No extension of time is believed to be necessary. If, however, an extension of time is required, the undersigned hereby authorizes the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

By /Amy J. Pattillo, Reg. No. 46,983/
AMY J. PATTILLO
Registration No. 46,983
P.O. BOX 161327
AUSTIN, TEXAS 78716
ATTORNEY FOR APPLICANTS
Telephone: 512-402-9820
Facsimile: 512-306-0417